Principles of accessibility design

According to W3C Web Content Accessibility Guidelines (WCAG) the four basic principles of accessibility design are perceivability, operability, understandability and robustness. These principles offer a foundation for creating software that is usable by people with impairments. The key guidelines under these principles are summarized below.

- Perceivability
- Operability
- Understandability
- Robustness

Perceivability

An application that is perceivable means that information can be identified by more than one sense. Some of the recommended practices to ensure perceivability are listed below.

Text alternatives [Recommended]

Text alternatives should describe non-text content such as labels and icons. Similarly, an application should provide captions and where possible audio to accommodate users with hearing and vision disabilities.

Captions for media content [Recommended]

Include captions for pre-recorded audio content. Offer alternative formats and transcripts for videos without subtitles. Add audio descriptions for videos with important visual information.

Color contrast [Recommended]

Ensure sufficient contrast between text and background colors, following WCAG guidelines. Avoid relying solely on color to convey information.

Resizable text [Recommended]

It should be possible to resize text without assistive technology without loss of content or functionality. This helps people with mild visual disabilities by allowing them to scale text and text-based controls on a web page, without requiring the use of assistive technology.

Clear error messages [Mandatory]

Provide clear and concise error messages with suggested solutions. Ensure error messages are programmatically linked to the corresponding form field. Offer tips or guidance for error correction where possible.

Descriptive titles [Mandatory]

Web pages/app screens should have titles that describe topic or purpose.This facilitates an easy and unambiguous identification of the webpage & also helps in a more relevant and visible presence in the search engine results.

Semantic HTML markup [Mandatory]

Use semantic HTML markup to structure content and provide meaningful information to assistive technologies. Proper use of headings, lists, and landmarks helps users navigate and understand the organization of content.

Operability

Software applications should allow users to perform all the actions necessary to navigate the user interface. Navigation should be seamless via all methods, such as the mouse, arrow keys and trackpad. When software is operable, it facilitates the use of assistive technologies like mouth sticks or speech recognition. Some of the recommended practices include:

Keyboard accessibility [Recommended]

Enable keyboard accessibility to ensure all elements are accessible through a keyboard. Prioritize keyboard access to form fields and interactive features. Allow using the "Enter" key as a substitute for clicking buttons.

Consistent and logical navigation [Mandatory]

Provide simple and consistent navigation. The default keyboard navigation order must be logical and intuitive. This generally means that it follows the visual flow of the page—left to right, top to bottom.

Focus indicators [Mandatory]

Provide focus indicators for interactive components so users can see what element has current keyboard focus.

Time limits [Mandatory]

Avoid problematic time limits for user interactions. Users should be provided with adequate time to provide responses without unnecessary time limits.

Define purpose of input fields [Mandatory]

The purpose of each input field collecting information about the user should be programmatically or described to enable use of assistive technology which can help users to complete forms.

Multiple navigation options [Recommended]

There should be more than one way available to locate a web page within a set of web pages except where the web page is the result of, or a step in, a process.Providing multiple navigation options can help people find information faster, particularly those with visual or cognitive impairments.

Search functionality [Mandatory]

Implement a robust search function that allows users to quickly find specific information or services by entering keywords or phrases. Supplementing navigation with search functionality provides an alternative method for users to access content and can be especially useful for users with specific information needs.

Easy access to important information [Recommended]

Ensure that important information, such as contact details, frequently asked questions, and key services, is easily accessible from any page of the website or system. Provide prominent links or buttons that direct users to essential information, reducing the time and effort required to find relevant content.

Understandability

Software that is understandable means that the user comprehends both the information it presents and the requirements for operation. Some of the practices to make software understandable include:

Use plain language [Mandatory]

Write content in plain language that is easy to understand for all users, regardless of their education level or background. Avoid technical jargon, complex terminology, and unnecessary verbosity to ensure clarity and comprehension.

Consistent format [Mandatory]

The application should be predictable, and its format should be consistent across pages.

Descriptive instructions [Mandatory]

All instructions and error messages should be descriptive and easy to follow, and software should help users correct their mistakes.

Limit reliance on sensory perception [Recommended]

Instructions provided for understanding and operating content should not rely solely on sensory characteristics of components such as shape, colour, size, visual location, orientation or sound. This is because some visually challenged users cannot perceive shape, size or colour. Alternative textual identification should be provided.

Multi language support[Recommended]

Where possible provide Kinyarwanda translations for software menus especially for software applications meant to be used by citizens.

Clear reading sequence [Recommended]

When the sequence in which content is presented affects its meaning, a correct reading sequence should be programmatically determined. This will enable assistive technologies such as screen readers to read the content in the correct order.

Logical structure for content [Mandatory]

Structure content in a logical and intuitive manner, using headings, lists, and clear signposts to help users navigate and find information quickly. Organizing content hierarchically improves readability and facilitates scanning and comprehension.

Robustness

Robustness means that software should be compatible with most devices and assistive technologies. Robust software adapts to ensure content is accessible as technologies evolve. Key practices include:

Responsive design [Mandatory]

Ensure that government websites and systems are responsive and adaptable to different screen sizes and devices. Responsive design enhances accessibility by providing a consistent user experience across desktops, tablets, and mobile devices.

Compatibility with assistive technologies [Recommended]

Ensure compatibility with various assistive technologies and assistive devices.Regularly test the software with a variety of assistive technologies, including screen readers, screen magnifiers, and voice recognition software, to identify and address accessibility issues.

Common standards [Mandatory]

Use common web technologies and adhere to standards and rules.Use consistent markup and standardized UI components.

Government institutions should also refer to the latest version of the Web Content Accessibility Guidelines (WCAG) which is a set of recommendations by W3C for web accessibility. WCAG offers guidelines and success criteria for making web content perceivable, operable, understandable, and robust for users with impairments